CS 7545: Machine Learning Theory	**Fall 2021**
Lecture 16	
Name: Tyler LaBonte	**Professor:** Santosh S. Vempala

1 Support Vector Machines

In certain learning scenarios, the data may not be separable. In that case we will have to minimize a loss function. One reasonable choice is the hinge loss: we'd like to find $\boldsymbol{w}$ such that for $\epsilon_i \geq 0$,

$$\boldsymbol{w}^\top \boldsymbol{x}_i \begin{cases} \geq 1 - \epsilon_i & \ell(\boldsymbol{x}_i) = 1, \\ \leq -1 + \epsilon_i & \ell(\boldsymbol{x}_i) = 0. \end{cases} \tag{1}$$

We will minimize the hinge loss $\sum_i \epsilon_i$. Note that if the data is separable, all the $\epsilon_i = 0$. Recall

$$\gamma = \min_{\boldsymbol{x}} \frac{|\boldsymbol{w}^{\star\top} \boldsymbol{x}|}{\|\boldsymbol{w}^\star\|}. \tag{2}$$

So if $\|\boldsymbol{w}^\star\| = 1$ and $\|\boldsymbol{x}\| \leq 1$,

$$\|\boldsymbol{w}^\star\| = \frac{\min_{\boldsymbol{x}} |\boldsymbol{w}^{\star\top} \boldsymbol{x}|}{\gamma} = \frac{1}{\gamma}. \tag{3}$$

This shows that to obtain a balance between a good margin and good classification, we can add a norm constraint to the objective function:

$$\min \|\boldsymbol{w}\|^2 + c \sum_i \epsilon_i. \tag{4}$$

This is called a support vector machine (SVM). We can adjust c based on the data and application. SVM requires solving a quadratic program, but we can use perceptron to do something similar.

Theorem 1.1: SVM Perceptron

The number of mistakes of perceptron is at most $\min_{\boldsymbol{w}} \left(\frac{1}{\gamma_{\boldsymbol{w}}^2} + 2\text{Hinge}(\boldsymbol{w}) \right)$. So, perceptron is like solving SVM with $c = 2$.

Proof: On a mistake, $\boldsymbol{w} \cdot \boldsymbol{w}^\star \leftarrow \boldsymbol{w} \cdot \boldsymbol{w}^\star + \ell(\boldsymbol{x}_i)(\boldsymbol{w}^\star \boldsymbol{x}_i)$, an increase of at least $1 - \epsilon_i$. Let $L = \sum_i \epsilon_i$, then after M mistakes,

$$|\boldsymbol{w} \cdot \boldsymbol{w}| \geq M - \sum_{\text{mistakes } i} \epsilon_i \geq M - L. \tag{5}$$

And,

$$\boldsymbol{w}^\top \boldsymbol{w} \leftarrow \boldsymbol{w}^\top \boldsymbol{w} + \|\boldsymbol{x}\|^2 + 2\ell(\boldsymbol{x})(\boldsymbol{w}^\top \boldsymbol{x}). \tag{6}$$

Since $\|\boldsymbol{x}\| \leq 1$ and $\ell(\boldsymbol{x})(\boldsymbol{w}^\top \boldsymbol{x}) \leq 0$ (mistake) we have that after M mistakes, $\boldsymbol{w}^\top \boldsymbol{w} = \|\boldsymbol{w}\|^2 \leq M$. By Cauchy-Schwarz, $|\boldsymbol{w} \cdot \boldsymbol{w}^\star| \leq \|\boldsymbol{w}\|\|\boldsymbol{w}^\star\|$, so $M - L \leq \sqrt{M}\|\boldsymbol{w}^\star\|$. Then since $\|\boldsymbol{w}^\star\|^2 = \frac{1}{\gamma^2}$, we can square both sides to obtain

$$M^2 + L^2 + 2LM \leq \frac{M}{\gamma^2} \implies M \leq \frac{1}{\gamma^2} + 2L - \frac{L^2}{M}. \tag{7}$$

We can drop the last term to obtain the answer. ■

We could also add a learning rate in front of $\ell(\boldsymbol{x})\boldsymbol{x}$ to achieve a constant other than 2.

2 Random Projection

One question is whether we can speed up halfspace learning by working in lower dimension. This is possible via random projection. Suppose $\mathbf{R}$ is a $d \times k$ matrix where $R_{ij} = \frac{1}{\sqrt{k}}\mathcal{N}(0,1)$. We can map $\boldsymbol{x} \in \mathbb{R}^d$ to $\boldsymbol{x}' \in \mathbb{R}^k$ by $\boldsymbol{x}' = \mathbf{R}^\top \boldsymbol{x}$. Clearly,

$$\mathbb{E}[\|\mathbf{R}^\top \boldsymbol{x}\|^2] = \|\boldsymbol{x}\|^2. \tag{8}$$

We will use the following versions of the Johnson-Lindenstrauss Lemma:

Theorem 2.1: Johnson-Lindenstrauss

$$\Pr[|\|\boldsymbol{x}'\|^2 - \|\boldsymbol{x}\|^2| \geq \epsilon\|\boldsymbol{x}\|^2] \leq 2\exp\left(\frac{-(\epsilon^2 - \epsilon^3)k}{4}\right), \tag{9}$$

$$\Pr[|\boldsymbol{x}' \cdot \boldsymbol{y}' - \boldsymbol{x} \cdot \boldsymbol{y}| \geq \epsilon\|\boldsymbol{x}\|\|\boldsymbol{y}\|] \leq 2\exp\left(-c\epsilon^2 k\right). \tag{10}$$

So, $k = \mathcal{O}\left(\frac{\log m}{\epsilon^2}\right)$ is sufficient to preserve m vector lengths and pairwise distances to within ϵ relative error and inner products to within ϵ additive error.

If $\epsilon = \gamma/2$, then we need $\mathcal{O}\left(\frac{k}{\epsilon}\log\frac{1}{\epsilon} + \frac{1}{\epsilon}\log\frac{1}{\delta}\right)$ samples to PAC learn. So, overall, we can project to dimension $\mathcal{O}\left(\frac{1}{\gamma^2}\log\frac{1}{\delta\epsilon}\right)$ to solve halfspace learning much faster.

This theorem also shows that margin constraints restrict the class of halfspaces to a lower complexity. In particular, the VC-dimension of halfspaces with margin γ is $\mathcal{O}\left(\frac{1}{\gamma^2}\log\frac{1}{\gamma}\right)$.